User's manual, JED AVR595 SBC using Atmel's ATmega2561

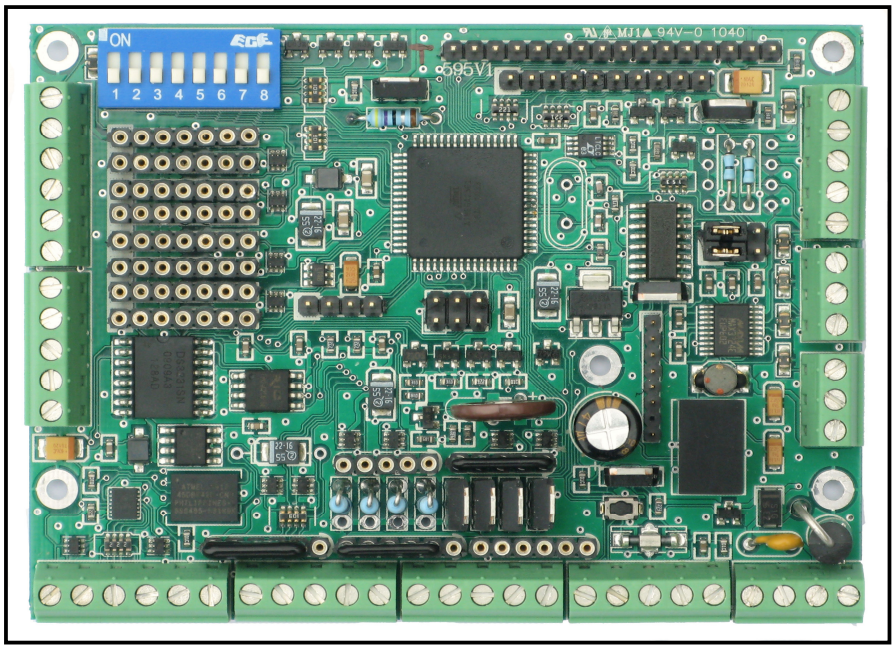
(Luke De Vincentis, Dec 2nd 2010)

The AVR595 board is a small form factor, general-purpose SBC from JED. It has been designed specifically with field-based low power data logging applications in mind. It has a powerful 8 bit AVR CPU and features a highly accurate RTC, loads of onboard flash memory, very high resolution 24 bit ADC, direct LCD and keypad support as well as multiple digital/analog CPU inputs and a number of high current sinking FET driven outputs. If you are looking for a compact, highly versatile, industrial strength SBC then the AVR595 is a great choice!

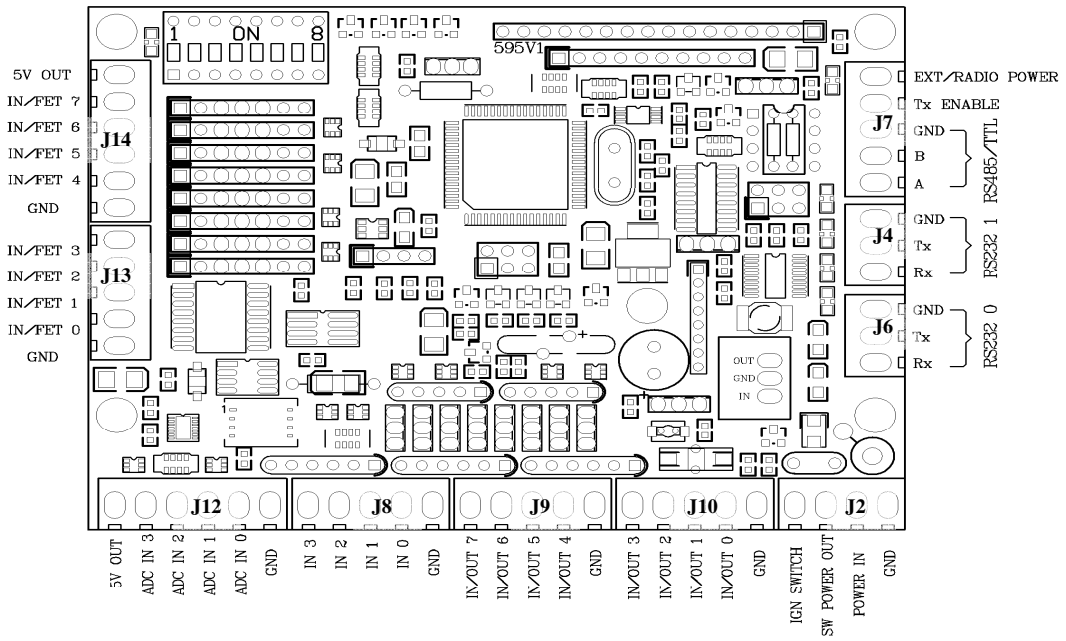
At the heart of the AVR595 is the ATmega2561 CPU from Atmel. It features the standard ATmega core CPU with 8K RAM, 256K FLASH and 4K of EEPROM all in a 64 pin package. It has 2 serial UARTs, I^2C and SPI interfaces, and 54 I/O lines. The full device data-sheet is 449 pages long, and is available at:
http://www.atmel.com/dyn/products/product_card.asp?family_id=607&family_name=AVR+8%2DBit+RISC+&part_id=3632

This manual provides a basic quick start information guide followed by a comprehensive step by step tour of the AVR595 schematic covering all board functions.

In the following manual notes, reference is made repeatedly to CPU functions accessed by port number, and the easiest way to find out what is inside the CPU "behind" the pins connected to the outside world on the AVR595 board is simply to open the Atmega2561 data sheet and use the CTRL F (Find) function to search the PDF data sheet by port number in the CPU data. (The format used for port addresses is "PB3" for Port B, bit 3, in both this manual, in the Atmel data sheet, in the JED schematic diagrams, so all documents are searchable with the same "P/letter/bit-number" format in the search.)



Quickstart Guide



To get up and running with the AVR595 board, on J2 link the 'IGN SWITCH' input to the 'Power In' terminal then apply 6.5-30v DC to the 'POWER IN and 'GND' terminals. The board comes loaded with a demo C program which displays a test menu on a terminal running at 9600 baud (8,N,1) via RS232 0 on J6. This demo program allows the user to test all functions on the board. Un-programmed the board draws under 30mA in quiescent current.

A WALK THROUGH THE SCHEMATIC, FUNCTION BY FUNCTION, PAGE BY PAGE ..4

A walk through the schematic, function by function, page by page.

CPU basics, Power Supply, Voltage Reference (Schematic sheet 1)

On Sheet 1 the CPU power and oscillators are covered.

Crystal
XTAL1 is normally a 3.6864 Mhz, but is deliberately NOT surface mounted so JED or users can change this to any frequency up to 16.0 Mhz. This is loaded as the default CPU clock source.

Oscillator
The **LTC6930** is a high precision, low power oscillator from Linear Tech. This can be loaded as an alternative to the standard crystal for a high precision, ultra low power 3.6864 Mhz CPU clock source.
See: http://cds.linear.com/docs/Datasheet/6930fb.pdf

CPU Digital power and Analog power
Power is provided to the CPU via EMC filtering inductors LC2 and LC1. Many bypass capacitors, a mix of low ESR Tantalum and ceramic, are used to bypass all the power pins and the voltage reference to the ground plane on an inner layer of the board.

Voltage reference
An accurate, 4.096 volt reference (U2, Linear Tech LT1790ACS6-4.096,) feeds the CPU's AREF pin 62. This part does not need trimming, as it has a basic accuracy of 0.05% and a drift less than 10ppm/deg. C over the full industrial temperature range.

In a vehicle-based application, the board runs from 12 volt ("always on") vehicle power, and is transient protected with a polyfuse and transorb for load-dump or reverse polarity and filtered. (This filtered power is available on a screw terminal to run off-board equipment, eg a modem or radio. The filtering can be extended off-board, or an external Li battery or NiMH battery backup with charger can also be linked in here.

Power can be switched on by the RTC alarm using the RTC interrupt output. The board will run from as low as 6.5 volts, but is fully protected for load dump and transients or reverse voltages for 12 volt automotive applications.

A Recom switching regulator with a 6.5 to 34 volt input range and 5 volts out at 0.5A (18v max) is used as the main onboard 5volt supply. The 5 volt main switching regulator can also power the LCD backlight, if used.
(See: http://www.recom-international.com/pdf/innoline/R-78xx-0.5.pdf)

Vcc at 5.0 volts is available to outside devices on J14.6 (or this board could have no regulators, and be powered completely from external regulators or a battery-backed power system can feed into these terminals, in low power logger applications.)

Vcc at 3.3V volts is regulated on board with a small linear regulator to run the DataFlash.

The power supply control and regulator components provide the following:

- Low-power application suited (eg a vehicle-based data logger), it has a 12 volt "ignition" sense input which powers up a system when the vehicle key is turned. Once the board is powered up, CPU "Hold on" signal via **PB4** allows the board to remain powered up once ignition signal is removed.

- "Vin-switched" is available on a screw terminal on J2.2 to power off-board radio/modems.

In a battery-powered and/or solar-powered data logging system, the user can either run the CPU all the time and power it down internally with the flexible power control register, or close the CPU down and have the RTC power it up at regular intervals.

LCD display and DIP Switch (Schematic Sheet 2)

Port C is pinned out to J15 for the LCD. The LCD can be a 4 by 20, a 2 by 40 or a 4 by 40 text display (using two enable lines). The LCD text display connects to J15. This connector matches to pin order of common HM44780 display pinouts, and the port C pins **PC4 … 7** connect to the LCD display's D4 … 7 data lines. The display is used in "nibble" mode with two 4-bit transfers per byte, as is provided by all compiler LCD drivers. Port **PC0** is the Register Select pin, **PC1** is R/W*, and **PC2** is the first LCD E1 (enable) pin. **PB7** provides LCD bias control via PWM output **OC1C**.

A flexible backlight powering system is provided, which allows the CPU to control the power on/off for the LCD backlight. An N-channel power FET, F8 is controlled by port **PC3,** and this grounds the –ve end of the backlight LED array. The positive end of the LED array is fed to Vcc.

Keyboard, DIP switch, IO expansion (Schematic Sheet 3)

Keyboard

Located on J3 this can be a 4 by 4 matrix or 8 individual switches to a ground pin on the keyboard connector. The user can configure the keyboard style by controlling the data direction registers associated with the ports **PA0 … PA7**. Software in a number of compilers can automatically scan at least a 4 x 4 keyboard on one port (Port A's 8 pins, 4 as inputs and 4 as outputs).
Alternatively an 8 way DIP switch is also provided for use on Port A as a general purpose onboard user input to select various operating modes etc.

The ATmega2560 CPU includes a powerful hardware implementation of "SPI", a high-speed clocked synchronous communications protocol (also compatible also with National's Microwire). Lines MOSI (Master Out Slave In), SCK (clock) and MISO (Master In Slave Out) use in turn CPU port bits **PB2**, **PB1** and **PB3**. Each device communicating with the CPU via SPI needs a CPU pin allocated as a LOW true chip select.

As well, the SPI system on this board is used to communicate with the on-board DataFlash memory (using port **PG5** as the low-true chip select)

CPU Flash programming

The USART0 system and SPI clock is used for CPU FLASH memory programming, and J1, a 6-pin connector on the board has the first six pins allocated to device programming, in a way compatible with Atmel's ISP programmer's 6-pin ribbon cable connector. Ground, 5volt Vcc and RESET* lines are on this connector as well. (See: http://www.atmel.com/dyn/products/tools_card.asp?family_id=607&family_name=AVR+8%2DBit+RISC+&tool_id=3808)

RS232 Ser0 and Ser1, RS485, V-Drive (Schematic Sheet 4)

The two serial interfaces are interfaced to RS232 user lines with a RS232 transceiver MAX 3224ECUP, U8.

Serial0 interface TX0/RX0 is available on J6. Ser0 is multiplexed with the V-Drive via U9. The control pin for this is USB_ON (CPU pin **PG3**) which at logic high selects SER0 and at logic low selects the V-drive and supplies power to it via F6.
Serial1 interface TX1/RX1 is available on J4. Ser1 is also shared with the RS485/TTL option via jumpers on L1. An LTC485 tranceiver may also be loaded upon request to provide half duplex RS485 communications via J7. The Tx enable is available externally via J7.2 or under CPU control via pin **PGO**. As standard the Ser1 option features the TTL configuration with link resistors in R22 and R23 in place of the RS485 transceiver for TTL signals for direct connection to a radio/modem. J7.1 also provides a switched 5v filtered power output for powering an external radio, modem etc. This may be controlled via the RADIO_ON signal on pin **PG1**.

Jumper settings for Ser1 are as follows:

RS232	Link L1 1-3, 2-4
RS485/TTL	Link L1 3-5, 4-6

V-Drive USB
A USB memory stick interface option is available for logging/file transfer, via a VDRIVE2 USB host system which can be mounted on a bracket accessible externally. A flying cable connects from the USB host card to the main CPU card via 2mm header J5. A serial port connects to the interface at CMOS levels, and both devices have simple ASCII or binary serial commands. Files can be created, opened, read or written and directories can be created or deleted. Files on the USB stick are compatible with FAT file systems for Windows/DOS systems.
(See: http://www.ghielectronics.com/details.php?id=1&sid=3
and http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_VDRIVE2.pdf)
Upon request there is a separate document available from JED which details the initialization sequence and command set for the V-Drive.

I²C, RTC and memory (Schematic Sheet 5)

I²C expansion system
The ATmega2561 CPU includes a full hardware implementation of the I²C communications system for expanding applications hardware to other chips, either on-board or off-board. (Atmel calls it the TWI or Two Wire Interface).

On the AVR595 board, the I²C interface is used to communicate with the Real Time Clock (DS3232), and the FRAM non-volatile device (FM24C256).

The interface is available on J11, a 4-pin header for expansion off-board. This header includes a 5volt Vcc pin and ground pin, as well as the SDA and SDL signal lines which are both pulled up to Vcc via 2k7 resistors. The interface is non-buffered, and the pinout is compatible with a range of I²C expansion boards designed by JED in connection with the AVR200 project board. (see: http://www.jedmicro.com.au/avr200.htm) Interface boards available include eight FET outputs, eight logic inputs, an LCD interface to a range of displays, and a 4 x 4 keyboard scanner.

DS3231 Real Time Clock, Battery backed RAM and Temperature transducer
The DS3232 provides a Real-Time-Clock to the AVR595 system. It runs from the I²C bus, and as well as very accurate time and calendar functions, it provides an alarm capable of automatically powering up a sleeping system. The timekeeping functions and the RAM are battery backed by a rechargeable, lithium cell held in a secure, vertical battery holder. The 32Khz crystal is inside the moulded package, protecting it from mechanical damage and effects from electrical or mechanical noise to the high-impedance crystal pins, which are now completely internal.

This RTC device uses a temperature-compensated crystal oscillator (TXCO) to provide a time-keeping accuracy of +/- 2ppm from 0 to 40 degrees C and +/- 3.5ppm from –40 to +85 degrees C. (This later is equivalent to +/- 2 minutes per year.) The clock counts from seconds to years and includes leap-year compensation to year 2099. There are two alarm register sets with time and date comparators. Enable registers enable one or both alarm systems to drive the interrupt

output pin. This can be used to either power up a completely powered-down system via power supply control logic, or generate a PIN-CHANGE interrupt to a power-up CPU which is in idle mode.

The local temperature of the RTC chip can be read out of a register in the RTC. This is effectively the PCB temperature, as the RTC power dissipation is close to zero. The temperature accuracy is +/- 3 degrees Centigrade. Data is read out as a 9 bit number plus sign, with 0.25 degrees Centigrade resolution.

The RTC is powered by a 3v lithium rechargeable battery from Panasonic. This allows the RTC to continue operating when the main board supply is turned off and recharged via an onboard constant current charging circuit when the board is powered up. (See: http://datasheets.maxim-ic.com/en/ds/DS3232.pdf)

DataFlash 8MByte Flash memory
This chip (AT45DB642D-TU, U9) is accessed via the SPI bus and provides high-speed sector-erased data storage with sector-sized dual ram buffering on the chip. Port **PG5** is the DataFlash chip select (low true). This can be accessed at high speed as linear memory space for data logging storage, as look-up memory for industrial processes or can even be formatted as a FAT file system and used for conventional file storage applications, maybe in conjunction with a SD/MMC memory card, and files transferred by name between storage devices.

It can also hold replacement images of the main CPU Flash memory, loaded via a communications interface, checked, and then loaded into the CPU via a boot-loader for remote field updates of far-away systems.

(See http://www.atmel.com/products/DataFlash/ for data on the DataFlash devices);

FRAM non-volatile 32Kbyte memory
An FM24C256 or FM24C512, 32/64 Kbyte Ramtron FRAM (Ferroelectric Nonvolatile RAM (see (via I^2C), is provided for data buffering and non-volatile short-term storage during data logging. It has a 10 billion read/write cycle life (compared with 100,000 operations typical for EEPROM) and does not have the 10 ms erase and write times of an EEPROM. Typical applications are for full sector buffers for data being assembled prior to writing to DataFlash or USB memory stick. Data is generated by events as small records, and then written when a full sector has accumulated in the FRAM. This has speed and power consumption advantages in applications where data is needed in a steady stream, or needs to be logged with minimum power consumption.
(See: http://www.ramtron.com/lib/literature/datasheets/FM24C256ds_r3.1.pdf) The FRAM devices are an interesting technology. (See: http://www.ramtron.com/doc/AboutFRAM/Default.asp);

Eight Analog or Digital Inputs and Delta-Sigma ADC (Schematic sheet 6)

The Atmega2561 provides eight channels of analog input via CPU port **PF0 … 7** on J13 and J14. This has 10 bits conversion resolution, and various in-CPU modes and control register bits combining channels in differential modes, control speed of the conversion clock, controlling gains on some channels and can even close the CPU down into a 'quiet' mode for lower-noise conversions. (Twenty-two pages are devoted to the ADC in the chip manual.)

Inputs
The eight "analog" inputs all can have user-installed pull-down resistors (eg 200 Ohms to terminate 4-20 mA current sources), can have user-installed pull-up resistors to Vcc to provide a top resistor and hence, current to a resistive transducer with one end to ground, a user-installed series resistor for protection, and then an optional second pull-down resistor as the bottom end of the divider pair (working with the series resistor), to scale higher input voltage input ranges to a range of 0 to 4.096 volts. The analog range is 0 to 4.092 volts for a count of 0-3FFh. The configurations of these loaded resistors for gain, pull-up or pull-down are shown on sheet 6 of the schematic.

All pins of these "input" pin group can be used for digital inputs (read in parallel as CPU ports **PF0 … PF7**.)

(Alternatively, FETs can be loaded with Source to socket strip pin 3 (Gnd.), Drain to pin 4 (external screw terminal) and Gate to pin 5 (CPU) and these can become four groups of four general-purpose outputs. (If FETs are loaded, 4K7 resistors must be loaded between pins 6 and 7 of the strips associated with each line from the FET gates to Gnd.) See Sheets 3 for a loading diagram. (Not available on prototype as holes are too small).

Delta-Sigma ADC
The AVR595 is also equipped with a 4 input high resolution, 24 bit delta-sigma ADC chip from Linear Tech (LTC2492) See: http://cds.linear.com/docs/Datasheet/2492fd.pdf
This allows ultra fine resolution via a wide-range filtered micro-volt input signal direct conversion system which is ideal for thermocouples, etc. ADC filtered power is supplied via LC3 and bypass caps and ultra low leakage protection diodes are provided across all inputs. J12 provides access to the 4 ADC inputs as well as a filtered 5v output.

Four FET Outputs, 1-Wire and External Interrupts (Schematic sheet 7)

4 high current sinking FET outputs
These are on J9 via **PD6-PD7** and **PE2-PE3**.
Pull up or pull down resistor networks may be installed in L13 on these open drain outputs if desired.
Another 4 FET's may be added on J8 if required instead of the interrupts on **PB5-PB6** and digital inputs on **PD4-PD5**.
6 External Interrupts and 2 Digital Inputs
These are on CPU pins **PE4…PE7, PB5-PB6, PD6-PD7** and are available via J9 and J10 with 4k7 current limit resistors and 4v7 protection diodes. **PE4…PE7, and PB5-PB6** are also "pin change" interrupt inputs which can power up a sleeping system, and/or generate an interrupt on the rising or falling edge, or a "low" level. The **PB5** and **PB6** pins can also be used as output compare match pins for generating an external PWM signal.

1-Wire Interface
This interface uses connector J10.5 via CPU port pin **PE7**. R28 is installed if this option is desired and a pullup to Vcc is provided via the resistor network installed in L14.
(If not used as a 1-Wire interface, this pin can be used as a general-purpose I/O pin with pull up.)

CPU FLASH programming/re-programming

The AVR595 is programmed via the RX0, TX0, SCK and RESET* pins, which appear on a 6-pin header (J1) compatible with the Atmel's AVRISP MKII. This can program any part of the CPU FLASH memory, including the boot sectors. These are stocked by JED in Australia, and have a USB connection to the development PC. See: www.atmel.com/dyn/products/tools_card.asp?family_id=607&family_name=AVR+8%2DBit+RISC+&tool_id=3808

The CPU book block sector can be loaded with a loader program to enable this board's CPU FLASH to be reloaded from a section of the DataFlash memory on reboot, and as this memory can be loaded by the CPU over any communications or file transport system provided on the board, potentially the CPU could be re-programmed from:

- A file on a USM memory stick;
- A file transferred via any RS232 serial port;
- A file transferred via radio/GSM link

In each case, the user CPU program sets up the re-program sequence, communicating with its host or unloading a transferred file into the allocated DataFlash sectors (and checking it with a CRC checksum). Then CPU sets some flags in the CPU eeprom, and does a jump to the restart address. On startup in the boot loader, the system checks the eeprom flags, and if set, proceeds to erase the CPU FLASH memory and re-writes it from the file, again doing a CRC check. When re-loaded and checked, the CPU clears the eeprom flags, and restarts again, this time (as the flag is NOT set) the user program starts normally, "phones home" and tells the control centre a successful reload has occurred.

The user-written program would be responsible for transferring the file to the DataFlash nominated area in whatever way was convenient, but they would all use the same boot loader in the boot sector of the DataFlash, which JED would provide.

AVR programming languages.

Many different compilers are available for the AVR, and assembly language can even be used, via the free "Studio" software from Atmel's web site or supplied with the AVR ISP programmer on CD. Atmel list compiled language support from 16 different companies at: http://www.atmel.com/products/AVR/thirdparty.asp#compilers

At JED we have used the CodeVison C compiler for many years, and highly recommend it for large-scale projects as a very good implementation of ANSI C with good embedded libraries and good "code-wizards" for AVR hardware support (I^2C, SPI, 1-Wire, LCD, SD/MMC etc). See: http://www.hpinfotech.ro/

We have also used the BASCOM BASIC compiler from MCS Electronics, and found it a simpler system than the C, but easily picked up by beginners. It has built-in I^2C, SPI, UART, LCD, 1-Wire support, simulator, etc. See http://www.mcselec.com/index.php?option=com_content&task=view&id=14&Itemid=41

There is also an AVR PASCAL compiler from mikroElectronica ... it seems easy to "pick up" and use. It has SD/MMC support built in, as well as I^2C, SPI, 1-Wire etc. See: http://www.mikroe.com/en/compilers/#avr),

There is also the free "gnu avr-gcc" tool-set, including the WinAVR IDE. See: http://www.avrfreaks.net/index.php?module=FreaksTools&func=viewItem&item_id=145

There are also a lot of code support products reviewed at the very popular dedicated AVR sites: http://www.avrfreaks.net/

http://www.atmel.com/products/avr/thirdparty.asp

A customer support disk is also supplied with the board which contains demo code written in C using the CodeVision C complier mentioned above. This code shows how to access all CPU functions in addition to things like setting up serial comms, doing ADC conversions, reading inputs and driving outputs, reading and writing to the dataflash memory and talking to the VDrive.